Attorney's Docket No. 042933/270197 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.:	10/690,656	Confirmation No.: 5860
Appellant(s):	Muhonen et al.	
Filed:	10/22/2003	
Art Unit:	2152	
Examiner:	Thomas J. Dailey	
Title:	SYSTEM AND ASSOCIATED TERMINAL, METHOD AND COMPUTER PROGRAM PRODUCT FOR CONTROLLING STORAGE OF CONTENT	

Customer No.: 00826

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences," filed March 17, 2010.

1. ***Real Party in Interest.***

The real party in interest in this appeal is Nokia Corporation, the assignee of the above-referenced patent application.

2. ***Related Appeals and Interferences.***

There are no related appeals and/or interferences involving this application or its subject matter.

3. ***Status of Claims.***

All of the pending claims, namely Claims 1-7, 9 and 11-39, stand rejected and are the subject of the present appeal. The remaining claims, namely Claims 8 and 10, have been cancelled.

4. ***Status of Amendments.***

Appellants filed an amendment to the claims in response to a final Official Action dated November 24, 2009. By an Advisory Action of February 18, 2010, the Examiner denied entry of the amendment. There are no other unentered amendments in this application.

5. ***Summary of Claimed Subject Matter.***

The claimed invention will now be summarized with references to passages of the specification and drawings. It should be understood, however, that the references are provided solely for explanatory purposes, and should not otherwise in and of themselves be taken to limit the scope of the claimed invention.

Independent Claim 1 recites an apparatus including a processor and a memory storing executable instructions that in response to execution by the processor cause the apparatus to at least perform a number of functions. Pat Appl., FIG. 2 (processor 32, memory 34); p. 10, ll. 10-19; and p. 25, l. 27 – p. 26, l. 19. As recited, the functions include receiving, from a terminal located remote from the apparatus, a status of at least one piece of content stored in memory of the terminal, where each piece of content is associated with parameters including a client expiration time and a deletion priority value. *Id.* at FIG. 6, block **108**; p. 16, l. 20 – p. 17, l. 4; and p. 19, l. 18 – p. 20, l. 4. As also recited, the functions include sending one or more instructions to the terminal based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control storage of the at least one piece of content in memory of the terminal. *Id.* at p. 21, l. 1 – p. 22, l. 12.

Depending from Claim 1, Claim 2 to recites that sending one or more instructions includes determining when memory of the terminal has sufficient storage capacity for at least one subsequent piece of content. *Id.* at FIG. 6, blocks **110, 112**; and p. 20, ll. 5-23. And as further recited, when memory does not have sufficient storage capacity, one or more instructions are sent to instruct at least one of the terminal or a user of the terminal to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory of the terminal. *Id.* at FIG. 6, blocks **118, 120**; p. 21, ll. 14-21; and p. 22, ll. 13-28.

Dependent Claim 3, depending from Claim 2, recites that sending one or more instructions to delete at least one piece of content includes determining a plurality of pieces of content having an exceeded client expiration time. *Id.* at FIG. 6, block **116**; and p. 21, ll. 1-13. As also recited, sending the instruction(s) includes identifying a piece of content having a highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, where the comparison excludes any piece of content without an exceeded client expiration time. *Id.* at FIG. 6, block **118**; and p. 21, ll. 14-21. And sending the instruction(s) includes sending one or more instructions to instruct the terminal to delete the identified piece of content. *Id.*

Further, depending from Claim 3, Claim 4 recites that a piece of content is repeatedly identified, and one or more instructions to instruct the terminal to delete the identified piece of content are repeatedly sent, until one of memory of the terminal has sufficient storage capacity for the at least one subsequent piece of content, or each piece of content having an exceeded client expiration time has been identified and deleted. *Id.* at p. 21, l. 21 – p. 22, l. 7.

Dependent Claim 5 (depending from Claim 4) recites that the memory stores executable instructions that in response to execution by the processor, when memory of the terminal does not have sufficient storage capacity for at least one subsequent piece of content and each piece of content having an exceeded client expiration time has been identified and deleted, cause the apparatus to further perform a number of functions. *Id.* at FIG. 6, block **120**; and p. 22, ll. 13-20. As recited, these functions include identifying at least one piece of content having a highest deletion priority value from a comparison between the deletion priority values of any pieces of content remaining in memory of the terminal. *Id.* at p. 22, ll. 20-26. And the functions include sending one or more instructions to instruct the terminal to delete the identified at least one piece of content. *Id.*

Dependent Claim 6 (depending from Claim 1) recites that the memory is configured to store at least one piece of content, and that the parameters further include a server expiration time. *Id.* at p. 16, ll. 20-26. And as further recited, the memory stores executable instructions that, in response to execution by the processor, cause the apparatus to further perform sending at least one piece of content to the terminal. *Id.* at FIG. 6, block **100**; and p. 18, ll. 1-10.

Dependent Claim 7 (depending from Claim 6) recites that the memory stores executable instructions that in response to execution by the processor cause the apparatus to further perform a number of functions. As recited, the functions include monitoring the server expiration time of the at least one piece of content in memory of the apparatus to determine when at least one piece of content has an exceeded server expiration time. *Id.* at FIG. 6, block **102**; and p. 18, ll. 11-26. And the functions include deleting the at least one piece of content having an expired server expiration time when at least one piece of content has an exceeded server expiration time. *Id.* at FIG. 6, block **104**; and p. 18, l. 27 – p. 19, l. 7.

Independent Claim 12 recites an apparatus including a processor and a memory storing executable instructions that in response to execution by the processor cause the apparatus to at least perform a number of functions. Pat Appl., FIG. 2 (processor 32, memory 34); p. 10, ll. 10-19; and p. 25, l. 27 – p. 26, l. 19. As recited, the functions include sending, to another apparatus located remote from the apparatus, a status of at least one piece of content stored in memory of the apparatus, where each piece of content is associated with parameters including a client expiration time and a deletion priority value. *Id.* at FIG. 6, block **108**; p. 16, l. 20 – p. 17, l. 4; and p. 19, l. 18 – p. 20, l. 4. As also recited, the functions include receiving one or more instructions from the other apparatus based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control storage of the at least one piece of content in memory of the apparatus. *Id.* at p. 21, l. 1 – p. 22, l. 12.

Depending from Claim 12, Claim 13 recites that receiving one or more instructions includes receiving one or more instructions to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory. *Id.* at FIG. 6, blocks **118**, **120**; p. 21, ll. 14-21; and p. 22, ll. 13-28. And the one or more instructions are received when, based on a determination of when memory has sufficient storage capacity for at least one subsequent piece of content, the memory does not have sufficient storage capacity. *Id.*

Dependent Claim 14, depending from Claim 13, recites that sending a status comprises sending a status of the at least one piece of content to enable the other apparatus to determine

when at least one piece of content has an exceeded client expiration time. *Id.* at FIG. 6, block **116**; and p. 21, ll. 1-13. When the other apparatus determines a plurality of pieces of content have an exceeded client expiration time, receiving one or more instructions includes receiving one or more instructions to delete a piece of content having a highest deletion priority value from the respective plurality of pieces of content. *Id.* at FIG. 6, block **118**; and p. 21, ll. 14-21. In this instance, the respective piece of content has been identified by the other apparatus as the piece of content having the highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, where the comparison excludes any piece of content without an exceeded client expiration time. *Id.*

Further, depending from Claim 14, Claim 15 recites that, when the other apparatus determines a plurality of pieces of content have an exceeded client expiration time, instruction(s) to delete a piece of content having a highest deletion priority value are repeatedly received until one of memory of the apparatus has sufficient storage capacity for the at least one subsequent piece of content, or each of the respective plurality of pieces of content has been identified and deleted. *Id.* at p. 21, l. 21 – p. 22, l. 7.

Dependent Claim 16 (depending from Claim 15) recites that when the other apparatus determines a plurality of pieces of content have an exceeded client expiration time, and when the memory does not have sufficient storage capacity for at least one subsequent piece of content and each of the respective plurality of pieces of content has been identified and deleted, receiving one or more instructions includes a number of functions. *Id.* at FIG. 6, block **120**; and p. 22, ll. 13-20. As recited, these functions include receiving one or more instructions to delete at least one piece of content having a highest deletion priority value from any pieces of content remaining in memory of the apparatus. *Id.* at p. 22, ll. 20-26. In this regard, the respective piece(s) of content are those that have been identified by the other apparatus as the piece(s) of content having the highest deletion priority value from a comparison between the deletion priority values of the pieces of content remaining in memory of the apparatus. *Id.*

Independent Claim 19 recites a method of controlling storage of content in memory. As recited, the method includes receiving, at a network entity from a terminal located remote from the network entity, a status of at least one piece of content stored in memory of the terminal,

where each piece of content is associated with parameters including a client expiration time and a deletion priority value. Pat. Appl., FIG. 6, block **108**; p. 16, l. 20 – p. 17, l. 4; and p. 19, l. 18 – p. 20, l. 4. As also recited, the method includes sending one or more instructions from the network entity to the terminal based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control, from the network entity, storage of content in memory of the terminal. *Id.* at p. 21, l. 1 – p. 22, l. 12.

Depending from Claim 19, Claim 20 recites that sending one or more instructions includes determining when memory of the terminal has sufficient storage capacity for at least one subsequent piece of content. *Id.* at FIG. 6, blocks **110**, **112**; and p. 20, ll. 5-23. And as further recited, when memory does not have sufficient storage capacity, one or more instructions are sent to instruct at least one of the terminal or a user of the terminal to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory of the terminal. *Id.* at FIG. 6, blocks **118**, **120**; p. 21, ll. 14-21; and p. 22, ll. 13-28.

Dependent Claim 21, depending from Claim 20, recites that sending one or more instructions to delete at least one piece of content includes determining a plurality of pieces of content having an exceeded client expiration time. *Id.* at FIG. 6, block **116**; and p. 21, ll. 1-13. As also recited, sending the instruction(s) includes identifying, and thereafter sending one or more instructions to delete, a piece of content having a highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time. *Id.* at FIG. 6, block **118**; and p. 21, ll. 14-21. In this regard, the comparison excludes any piece of content without an exceeded client expiration time. *Id.*

Further, depending from Claim 12, Claim 22 recites that a piece of content is repeatedly identified, and one or more instructions to instruct the terminal to delete the identified piece of content are repeatedly sent, until one of memory of the terminal has sufficient storage capacity for the at least one subsequent piece of content, or each piece of content having an exceeded client expiration time has been identified and deleted. *Id.* at p. 21, l. 21 – p. 22, l. 7.

Dependent Claim 23 (depending from Claim 22) recites that when memory of the terminal does not have sufficient storage capacity for at least one subsequent piece of content

and each piece of content having an exceeded client expiration time has been identified and deleted, the method further comprises a number of functions. *Id.* at FIG. 6, block **120**; and p. 22, ll. 13-20. As recited, these functions include identifying, and thereafter sending one or more instructions to delete, a piece of content having a highest deletion priority value from a comparison between the deletion priority values of any pieces of content remaining in memory of the terminal. *Id.* at p. 22, ll. 20-26.

Dependent Claim 24 (depending from Claim 19) recites that the method further includes receiving at least one piece of content at the network entity. *Id.* at FIG. 6, block **100**; and p. 18, ll. 1-10. And as further recited, the method includes sending at least one piece of content to the terminal such that the terminal receives, and thereafter stores, the at least one piece of content sent thereto. *Id.*

Dependent Claim 25 (depending from Claim 24) recites that the parameters further include a server expiration time, and that the method further includes a number of functions. *Id.* at p. 16, ll. 20-26. As recited, the functions include monitoring the server expiration time of the at least one piece of content in memory of the network entity to determine when at least one piece of content has an exceeded server expiration time. *Id.* at FIG. 6, block **102**; and p. 18, ll. 11-26. And the functions include deleting the at least one piece of content having an expired server expiration time when at least one piece of content has an exceeded server expiration time. *Id.* at FIG. 6, block **104**; and p. 18, l. 27 – p. 19, l. 7.

Independent Claim 29 recites a computer program product for controlling storage of content in memory, where the computer program product includes a computer-readable storage medium having computer-readable program code portions stored therein that in response to execution by a processor cause an apparatus to at least perform a number of functions. Pat Appl., FIG. 2 (processor 32, memory 34); p. 10, ll. 10-19; and p. 25, l. 27 – p. 26, l. 19. As recited, the functions include receiving, from a terminal located remote from the the apparatus, a status of at least one piece of content stored in memory of the terminal, where each piece of content is associated with parameters including a client expiration time and a deletion priority value. *Id.* at FIG. 6, block **108**; p. 16, l. 20 – p. 17, l. 4; and p. 19, l. 18 – p. 20, l. 4. As also recited, the functions include sending one or more instructions from the apparatus to the terminal

based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control, from the apparatus, storage of content in memory of the terminal. *Id.* at p. 21, l. 1 – p. 22, l. 12.

Depending from Claim 29, Claim 30 recites that sending one or more instructions includes determining when memory of the terminal has sufficient storage capacity for at least one subsequent piece of content. *Id.* at FIG. 6, blocks **110**, **112**; and p. 20, ll. 5-23. And as further recited, when memory does not have sufficient storage capacity, one or more instructions are sent to instruct at least one of the terminal or a user of the terminal to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory of the terminal. *Id.* at FIG. 6, blocks **118**, **120**; p. 21, ll. 14-21; and p. 22, ll. 13-28.

Dependent Claim 31, depending from Claim 30, recites that sending one or more instructions to delete at least one piece of content includes determining a plurality of pieces of content having an exceeded client expiration time. *Id.* at FIG. 6, block **116**; and p. 21, ll. 1-13. As also recited, sending the instruction(s) includes identifying a piece of content having a highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, where the comparison excludes any piece of content without an exceeded client expiration time. *Id.* at FIG. 6, block **118**; and p. 21, ll. 14-21. And sending the instruction(s) includes sending one or more instructions to instruct the terminal to delete the identified piece of content. *Id.*

Further, depending from Claim 31, Claim 32 recites that a piece of content is repeatedly identified, and one or more instructions to instruct the terminal to delete the identified piece of content are repeatedly sent, until one of memory of the terminal has sufficient storage capacity for the at least one subsequent piece of content, or each piece of content having an exceeded client expiration time has been identified and deleted. *Id.* at p. 21, l. 21 – p. 22, l. 7.

Dependent Claim 33 (depending from Claim 32) recites that the computer-readable storage medium has computer-readable program code portions stored therein that in response to execution by the processor, when memory of the terminal does not have sufficient storage capacity for at least one subsequent piece of content and each piece of content having an

exceeded client expiration time has been identified and deleted, cause an apparatus to further perform a number of functions. *Id.* at FIG. 6, block **120**; and p. 22, ll. 13-20. As recited, these functions include identifying, and thereafter sending one or more instructions to instruct the terminal to delete, a piece of content having a highest deletion priority value from a comparison between the deletion priority values of any pieces of content remaining in memory of the terminal. *Id.* at p. 22, ll. 20-26.

Dependent Claim 34 (depending from Claim 30) recites that the computer-readable storage medium has computer-readable program code portions stored therein that, in response to execution by the processor, cause the apparatus to further perform a number of functions. As recited, the functions include receiving at least one piece of content at the apparatus. *Id.* at FIG. 6, block **100**; and p. 18, ll. 1-10. And as further recited, the functions include sending at least one piece of content to the terminal such that the terminal receives, and thereafter stores, the at least one piece of content. *Id.*

Dependent Claim 35 (depending from Claim 34) recites that the parameters further include a server expiration time, and that the computer-readable storage medium has computer-readable program code portions stored therein that in response to execution by the processor, cause the apparatus to further perform a number of functions. *Id.* at p. 16, ll. 20-26. As recited, the functions include monitoring the server expiration time of the at least one piece of content in memory of the apparatus to determine when at least one piece of content has an exceeded server expiration time. *Id.* at FIG. 6, block **102**; and p. 18, ll. 11-26. And the functions include deleting the at least one piece of content having an expired server expiration time when at least one piece of content has an exceeded server expiration time. *Id.* at FIG. 6, block **104**; and p. 18, l. 27 – p. 19, l. 7.

Independent Claim 39 recites an apparartus that includes a means for storing at least one piece of content, where each piece of content is associated with parameters including a client expiration time and a deletion priority value. Pat. Appl., FIG. 6, block **108**; p. 16, l. 20 – p. 17, l. 4; and p. 19, l. 18 – p. 20, l. 4. As recited, the apparatus also includes a means for sending a status of the at least one piece of content stored by the apparatus to a network entity located remote from the apparatus. *Id.* And the apparatus includes a means for receiving one or more

instructions from the network entity based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control storage of the at least one piece of content by the apparatus. *Id.* at p. 21, l. 1 – p. 22, l. 12.

Appellants further note that independent Claim 39 provides an apparatus comprising a number of means-plus-function elements as permitted by 35 U.S.C. § 112, sixth paragraph. As explained on pages 25 and 26 of the present application, the steps or functions performed or capable of being performed by various apparatuses of embodiments of the present invention may be carried out by computer program instructions loaded onto a computer or other programmable apparatus, computer program instructions stored in computer-readable memory, or a special purpose hardware-based computer system, alone or in various combinations. Accordingly, to the extent independent Claim 39 recites means-plus-function elements, each of those elements may correspond to computer program instructions loaded onto a programmable apparatus or stored in computer-readable memory, or correspond to special purpose hardware.

6. ***Grounds of Rejection to be Reviewed on Appeal.***

Pending Claims 1, 6, 7, 9, 11-13, 17-20, 24-29 and 34-39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,130,908 to Pecus et al., in view of U.S. Patent No. 6,157,982 to Deo. And pending Claims 2-5, 14-16, 21-23 and 30-33 stand rejected as being unpatentable over Pecus, in view of Deo, and further in view of U.S. Patent Application Publication No. 2005/0172326 to Jerding et al. The remaining claims, namely Claims 8 and 10, have been cancelled.

7. ***Argument.***

As explained below, Appellants respectfully submit that the claimed invention is patentably distinct from Pecus, Deo and Jerding, taken individually or in any proper combination. In view of the remarks presented herein, Appellants respectfully request reconsideration and reversal of the rejections of all of the pending claims.

A. Claims 1, 6, 7, 9, 11-13, 17-20, 24-29 and 34-39 are Patentable

As indicated, the Official Action rejects Claims 1, 6, 7, 9, 11-13, 17-20, 24-29 and 34-39 as being unpatentable over Pecus, in view of Deo. In contrast to one aspect of the claimed invention, as reflected by independent Claim 12, Pecus does not explicitly or inherently (necessarily, if not explicitly) disclose an apparatus caused to direct transmission of a status of stored content to a remote apparatus, and receive instructions from the remote apparatus based upon the status and parameters associated with the content including a client expiration time and deletion priority value. The Examiner in the final Official Action concedes that Pecus does not disclose the aforementioned feature, but alleges that Deo discloses the feature and that it would have been obvious to one skilled in the art to modify Pecus per Deo to teach the apparatus of independent Claim 12. Appellants respectfully disagree, and maintain that even if one could argue Pecus and Deo disclose respective elements of independent Claim 12, there is no apparent reason for the combination of Pecus and Deo, and the Official Action does not provide sufficient reasoning for their combination.

As clearly explained by the Supreme Court in *KSR Int'l. Co. v. Teleflex, Inc.*, 127 S.Ct. 1727, 82 USPQ2d (BNA) 1385 (2007), any finding of obviousness should be based on an apparent reason to combine the prior art, and must be supported by more than mere conclusory statements. In the instant case, the Examiner includes the following explanation of Deo and the purported combination of Pecus and Deo:

> *17. ... Deo discloses sending one or more instructions from a processor to a remote terminal based upon the status of the content stored in memory to at least partially control storage at least one piece of content in memory of the terminal, said instruction including determining available memory capacity of the terminal and if said memory does not have sufficient storage capacity deleting content (column 3, lines 8-24, a computer (apparatus) remotely issues memory transactions (instructions) to a information device (terminal), those instructions being based upon the content of the information device's memory, and the computer (apparatus)* ***determines how much space is available as it has a map of the device memory in its own memory****).*
>
> *Thus, the combined teachings of Pecus and Deo would yield a system in which the memory management method of Pecus executed by the edge node (i.e. determining what entries are expired and which are marked for deletion) would be carried out by the NOC. Due to the fact, that Deo discloses a*

> *system in which a remote device memory transactions are controlled by another, separate device.*
>
> *Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Pecus and Deo in order to decrease the processing burden of a terminal that has less processing power available than a computer it is networked with (Deo, column 2, line 65-column 3, line 4).*

Official Action of Nov. 24, 2009, pp. 9-10 (emphasis added). Appellants disagree with this assessment of Pecus and Deo insofar as the combination is alleged to teach the claimed invention.

1. Pecus Fails to Disclose its Terminal having Less Processing Power

Even if the system of Deo does generally decrease the processing burden of a processing-limited terminal, the Official Action fails to make the connection between that general benefit of Deo and the particular system of Pecus. In this regard, the alleged benefit of Deo applies to processing-limited devices, but nowhere has the Official Action alleged or explained that the EN of Pecus is a processing-limited device that would benefit from a reduction in its processing burden to the extent that it actually would have been obvious to one skilled in the art to modify Pecus as alleged. In fact, in one passage, Pecus discloses its EN includes a controller that may be implemented by a general purpose computer or workstation, such as a PowerEdge server or Power Macintosh G4 server. *See* Pecus, col. 15, ll. 51-57. And in another passage, Pecus discloses that its EN may include "four dual-733 MHz Intel Pentium III processor servers (e.g., the Power Edge 2450 model server from Dell Computer Corp.), an RF gain amplifier, two satellite routers (e.g., the Enterprise1 from Harmonic Data Systems), a network switch (e.g., Model Catalyst 2924 from CISCO Systems), two remote power controllers (e.g., Model AP9211 from APC), a firewall device (e.g., model NetScreen-10 from NetScreen Technologies), multiport keyboard/display controller (e.g., model KVM-8 from APC), and keyboard/mouse/display unit." *Id.* at col. 14, ll. 31-56; and see FIG. 6. These disclosures, and the disclosure of Pecus in general, clearly runs contrary to any suggestion that a reduction in the processing burden on its EN is a concern that would lead one to modify the EN per Deo.

In response, the Examiner in the final Official Action notes that the test for obviousness is not whether the features of a reference may be bodily incorporated into the structure of another reference, but instead, what the combined teachings of the references would have suggested to one skilled in the art. Official Action of Nov. 24, 2009, p. 3. Even given this proposition, however, Appellants submit that "[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention." MPEP § 2141.02 (emphasis in original) (*citing W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540 (Fed. Cir. 1983). And considered as a whole, Pecus does not teach or suggest the type of processing-limited device to which Deo is directed, and therefore does not lend itself to the alleged motivation for modifying Pecus per Deo. In fact, the disclosure of Pecus in general clearly runs contrary to any suggestion that a reduction in the processing burden on its edge node is a concern that would lead one to modify the EN per Deo.

2. Deo's Solution Leads Away from the Alleged Combination

Briefly, Deo discloses a system and method for remotely managing memory in a portable information device from an external computer. As disclosed, the device memory is mapped into a portion of the computer memory to create a virtual device memory therein. To effectuate a change in the device memory, then, a user enters programming changes to be made to the information device. The programming changes alter the virtual device memory within the computer memory, and a memory manager resident in the computer determines what memory transactions are effective to alter the virtual device memory. The computer generates a serial stream of data indicative of memory transactions to effectuate a corresponding alteration of the device memory, and the data is transmitted to the information device to carry out the respective memory transactions and update the device memory.

As disclosed by Pecus its network operations center (NOC) may send a "RequestFileStatus" command to direct the edge node (EN) to provide the NOC with the status of locally-stored content, but the EN of Pecus makes decisions as to deleting content from its memory based on parameters including an expiration time and deletion priority value. In the system of Deo, on the other hand, the computer (analogous to the NOC of Pecus) makes

decisions as to deleting content from memory of the portable information device (analogous to the EN of Pecus). Instead of making those decisions based on any notification of content status from the portable information device, however, the computer maps the device's memory; thereby negating any need for the computer receiving, from the portable information device, the status of content stored in memory of that device. The Examiner in the final Official Action even cites this feature of Deo in application of Deo to independent Claim 12. Official Action of Nov. 24, 2009, pp. 9-10 (explaining that "the computer (apparatus) determines how much space is available [on the portable information device] as it has a map of the device memory in its own memory").

At best, then, one could argue that Deo reduces processing burden on its portable information device by mapping memory of that device at the computer, and having the computer control storage of content in memory of that device. But by adding these features (mapping memory of the terminal and controlling storage of content) to the NOC (computer) of Pecus, the EN (portable information device) no longer needs to notify the NOC of the status of content in its memory since the server will already have that information. Thus, it would not have been obvious to one skilled in the art to modify Pecus per Deo to realize an apparatus that receives, from a remote terminal, a status of content stored in memory of the terminal, and that prepares one or more instructions for transmission to the terminal based on that status to thereby control storage of content in memory of the terminal, similar to independent Claim 12. Instead, at best one may argue that the combination of Pecus and Deo yields an apparatus that maps the memory of another apparatus, and sends instructions to the other apparatus based on the map (and not a received status from the apparatus).

In response, the Examiner in the final Official Action again notes that the test for obviousness is not whether the features of a reference may be bodily incorporated into the structure of another reference, but instead, what the combined teachings of the references would have suggested to one skilled in the art. Official Action of Nov. 24, 2009, p. 4. Considered as a whole, Deo's solution explicitly includes mapping the memory of a portable information device at a the computer so as to reduce the memory management processing burden on the portable information device. And Appellants respectfully submit that the memory mapping feature of Deo may not be ignored when considering "what the combined teachings of the references would

have suggested to one skilled in the art," particularly when this feature enables the benefit of modifying Pecus per Deo alleged in the Official Action.

Appellants therefore respectfully submit that amended independent Claim 12, and by dependency Claims 13-18, is patentably distinct from Pecus. Amended independent Claims 1, 19, 29 and 39 recite subject matter similar to that of independent Claim 12, including the aforementioned controlling storage of content at a terminal based on multiple parameters associated with the content, and sending instructions from a remote network entity or apparatus to control storage of such content. Thus, Appellants also respectfully submit that amended independent Claims 1, 19, 29 and 39, and by dependency Claims 2-7, 9, 11, 20-28 and 30-38, are also patentably distinct from Pecus, taken individually or in any proper combination, for at least the reasons given above with respect to independent Claim 12.

In addition to the foregoing reasons, Appellants respectfully submit that various ones of dependent Claims 2-7, 9, 11, 13-18, 20-28 and 30-38 recite features further patentably distinct from Pecus and Deo, taken individually or in any proper combination. Examples of such dependent claims, including Claims 7, 25 and 35, are explained below.

3. Dependent Claims 7, 25 and 35

As to dependent Claims 7, 25 and 35, neither Pecus nor Deo, taken individually or in any proper combination, teach or suggest a server expiration time according to which content may be deleted from a network entity or apparatus (separate from the terminal with content associated with a client expiration time according to which locally-stored content may be deleted). The Official Action cites Pecus for allegedly disclosing this feature, with the Official Action alleging that the NOC of Pecus corresponds to the recited network entity. However, Appellants respectfully submit that nowhere does Pecus disclose its NOC monitors an expiration time (server expiration time) of its locally-stored content, and deletes content having an expired expiration time (server expiration time). Rather, Pecus only discloses deleting content locally-stored by its EN, and based upon a single expiration time (allegedly corresponding to the recited client expiration time). Thus, not only does Pecus not teach or suggest its alleged network entity deleting locally-stored content based upon monitoring an expiration time, Pecus does not teach

or suggest multiple expiration times associated with a piece of content. That is, Pecus does not teach or suggest both a client expiration time (from which content may be deleted from memory of a terminal), and a server expiration time (from which content may be deleted from the network entity that sends the content to the terminal), as recited by dependent Claims 7, 25 and 35.

Appellants note that in response to the foregoing, the Examiner merely repeats the Examiner's same position from a prior Official Action. Other than stating disagreement with Appellants' arguments, nowhere does the Examiner provide any explanation as to any point of disagreement with Appellants' arguments, or provide any other indication that the Examiner considered Appellant's arguments. As has been held, however, "[i]f a *prima facie* case [of obviousness] is made in the first instance, and if the applicant comes forward with reasonable rebuttal, whether buttressed by experiment, prior art references, or argument, the entire merits of the matter are to be reweighed." *In re Piasecki*, 745 F.2d 1468, 1472 (Fed. Cir. 1984).

B. Claims 2-5, 14-16, 21-23 and 30-33 are Patentable

The Official Action rejects Claims 2-5, 14-16, 21-23 and 30-33 as being unpatentable over Pecus, in view of Deo, and further in view of Jerding. As explained above, independent Claims 1, 12, 19, 29 and 39, and by dependency Claims 2-7, 9, 11, 13-18, 20-28 and 30-38, are patentably distinct from Pecus and Deo, taken individually or in any proper combination. Appellants respectfully submit that Jerding does not cure the deficiencies of Pecus and Deo. That is, even considering Jerding, none of Pecus, Deo or Jerding, taken individually or in any proper combination, teaches or suggests the aforementioned controlling storage of content at an apparatus based on multiple parameters associated with the content, and sending instructions from another, remote apparatus to control storage of such content, as per independent Claims 1, 12, 19, 29 and 39. Appellants therefore respectfully submit that independent Claims 1, 12, 19, 29 and 39, and by dependency Claims 2-7, 9, 11, 13-18, 20-28 and 30-38, are patentably distinct from Pecus, in view of Deo, and further in view of Jerding.

In addition to the foregoing reasons, Appellants respectfully submit that various ones of dependent Claims 2-7, 9, 11, 13-18, 20-28 and 30-38 recite features further patentably distinct

from Pecus and Deo, taken individually or in any proper combination. Examples of such dependent claims, including Claims 3-5, 14-16, 21-23 and 31-33, are explained below.

1. Dependent Claims 3, 4, 14, 15, 21, 22, 31 and 32

In contrast to dependent Claims 3, 14, 21 and 31, from which Claims 4, 15, 22 and 32 depend, nowhere does Pecus teach or suggest determining content having an exceeded client expiration time, and from that content, identifying content having the highest deletion priority value (thereby identifying content that is both expired and has the highest deletion priority value). That is, following the assertions of the Examiner, nowhere does Pecus teach or suggest determining expired content, and from that content, identifying content having the highest forced deletion flag (thereby identifying content that is both expired and has the highest forced deletion flag). Instead, Pecus treats its expiration time and forced deletion flag separate from one another in deciding whether to delete content. More particularly, Pecus discloses deleting all of the expired content or content marked for forced deletion; or first deleting content marked for forced deletion, and then expired content.

In response to the foregoing, the Examiner in the final Official Action states:

> *10. ... Pecus and Deo discloses, as substantially recited in the claims, determining a plurality of pieces of content having an exceeded client expiration time (Pecus, column 17, lines 15-20, "expired files" are identified), identifying a piece of content having a highest deletion priority value from a comparison of the deletion priority values of the pieces of content having an exceeded client expiration time (column 17, lines 20-24, the data manager checks for file(s) marked for forced deletion; i.e. a plurality of files' forced deletion flag is compared with the Boolean value "true" to determine if they should be deleted, "true" being the highest value for deletion priority; further, as all files are checked those that are expired will also be checked), and send one or more instructions instructing the terminal to delete the identified piece of content (Pecus, column 17, lines 15-28, if files are both expired and have are marked for forced deletion, they will be deleted).*

Official Action of Nov. 24, 2009, pp. 5-6. However, even if one could argue that the disclosed forced deletion flag of pieces of content of Pecus could correspond to the recited deletion priority values of pieces of content, Pecus does not teach or suggest any comparison between the flags of

the pieces of content, similar to the deletion priority values of Claims 3, 4, 14, 15, 21, 22, 31 and 32. The Examiner even concedes this point on page 15 of the Official Action.

The Examiner appears to suggest that even if Pecus treats its expiration time and forced deletion flag separate from one another in deciding whether to delete content, if content are both expired and are marked for forced deletion – and are thus deleted, this meets the feature of Claims 3, 4, 14, 15, 21, 22, 31 and 32. Appellants respectfully disagree and note that even if in this limited circumstance the end result of the claimed invention and Pecus overlap, the claims are not directed to the end result but instead the process by which that end result is achieved. Thus, even if the end results overlap, Pecus still fails to disclose the process by which the end result is achieved as recited by Claims 3, 4, 14, 15, 21, 22, 31 and 32.

2. Dependent Claims 5, 16, 23 and 33

Dependent Claims 5, 16, 23 and 33, which depend from respective ones of Claims 3, 14, 21 and 31 by way of respective ones of Claims 4, 15, 22 and 32, recite sending or receiving instruction(s) to delete content having an exceeded client expiration time, and from any remaining content, delete content having the highest deletion priority value, which is also absent from Pecus, Deo and Jerding, taken individually or in any proper combination. Pecus does disclose deleting content marked for forced deletion (alleged deletion priority value), and then expired content (alleged exceeding client expiration time) – i.e., first forced deletion, then expired. But even given this disclosure, Pecus does not teach deleting expired content, and then content having the highest deletion priority – i.e., first expired, then highest deletion priority value, similar to Claims 5, 16, 23 and 33.

The Examiner in the final Official Action responds to the foregoing in a manner similar to the above response presented for Claims 3, 4, 14, 15, 21, 22, 31 and 32. Again, however, even if one could argue that the disclosed forced deletion flag of pieces of content of Pecus could correspond to the recited deletion priority values of pieces of content, Pecus does not teach or suggest any comparison between the flags of the pieces of content, also similar to the deletion priority values of Claims 5, 16, 23 and 33.

8. ***Claims Appendix.***

The claims subject to this appeal are as follows:

1. (Previously Presented) An apparatus comprising a processor and a memory storing executable instructions that in response to execution by the processor cause the apparatus to at least perform the following:

receiving, from a terminal located remote from the apparatus, a status of at least one piece of content stored in memory of the terminal, wherein each piece of content is associated with parameters including a client expiration time and a deletion priority value; and

sending one or more instructions to the terminal based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control storage of the at least one piece of content in memory of the terminal.

2. (Previously Presented) The apparatus of Claim 1, wherein sending one or more instructions comprises:

determining when memory of the terminal has sufficient storage capacity for at least one subsequent piece of content; and when memory does not have sufficient storage capacity,

sending one or more instructions to instruct at least one of the terminal or a user of the terminal to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory of the terminal.

3. (Previously Presented) The apparatus of Claim 2, wherein sending one or more instructions to delete at least one piece of content comprises:

determining a plurality of pieces of content having an exceeded client expiration time;

identifying a piece of content having a highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, the comparison excluding any piece of content without an exceeded client expiration time; and

sending one or more instructions to instruct the terminal to delete the identified piece of content.

4. (Previously Presented) The apparatus of Claim 3, wherein identifying a piece of content, and sending one or more instructions to instruct the terminal to delete the identified piece of content, comprise repeatedly identifying a piece of content, and sending one or more instructions to instruct the terminal to delete the identified piece of content, until one of memory of the terminal has sufficient storage capacity for the at least one subsequent piece of content, or each piece of content having an exceeded client expiration time has been identified and deleted.

5. (Previously Presented) The apparatus of Claim 4, wherein the memory stores executable instructions that in response to execution by the processor, when memory of the terminal does not have sufficient storage capacity for at least one subsequent piece of content and each piece of content having an exceeded client expiration time has been identified and deleted, cause the apparatus to further perform the following:

identifying at least one piece of content having a highest deletion priority value from a comparison between the deletion priority values of any pieces of content remaining in memory of the terminal; and

sending one or more instructions to instruct the terminal to delete the identified at least one piece of content.

6. (Previously Presented) The apparatus of Claim 1, wherein the memory is configured to store at least one piece of content, wherein the parameters further include a server expiration time, and wherein the memory stores executable instructions that in response to execution by the processor cause the apparatus to further perform sending at least one piece of content to the terminal.

7. (Previously Presented) The apparatus of Claim 6, wherein the memory stores executable instructions that in response to execution by the processor cause the apparatus to further perform the following:

monitoring the server expiration time of the at least one piece of content in memory of the apparatus to determine when at least one piece of content has an exceeded server expiration time; and when at least one piece of content has an exceeded server expiration time,

deleting the at least one piece of content having an expired server expiration time.

8. (Cancelled)

9. (Previously Presented) The apparatus of Claim 1, wherein the each piece of content stored in memory of the terminal is associated with respective parameters.

10. (Cancelled)

11. (Previously Presented) The apparatus of Claim 9, wherein the memory stores executable instructions that in response to execution by the processor cause the apparatus to further perform the following:

associating each piece of content stored in memory of the terminal with respective parameters.

12. (Previously Presented) An apparatus comprising a processor and a memory storing executable instructions that in response to execution by the processor cause the apparatus to at least perform the following:

sending, to another apparatus located remote from the apparatus, a status of at least one piece of content stored in memory of the apparatus, each piece of content being associated with parameters including a client expiration time and a deletion priority value; and

receiving one or more instructions from the other apparatus based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control storage of the at least one piece of content in memory of the apparatus.

13. (Previously Presented) The apparatus of Claim 12, wherein receiving one or more instructions comprises receiving one or more instructions to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory, the one or more instructions being received when, based on a determination of when memory has sufficient storage capacity for at least one subsequent piece of content, the memory does not have sufficient storage capacity.

14. (Previously Presented) The apparatus of Claim 13, wherein sending a status comprises sending a status of the at least one piece of content to enable the other apparatus to determine when at least one piece of content has an exceeded client expiration time, and wherein, when the other apparatus determines a plurality of pieces of content have an exceeded client expiration time, receiving one or more instructions comprises receiving one or more instructions to delete a piece of content having a highest deletion priority value from the respective plurality of pieces of content, the respective piece of content having been identified by the other apparatus as the piece of content having the highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, the comparison excluding any piece of content without an exceeded client expiration time.

15. (Previously Presented) The apparatus of Claim 14, wherein, when the other apparatus determines a plurality of pieces of content have an exceeded client expiration time, receiving one or more instructions comprises repeatedly receiving one or more instructions to delete a piece of content having a highest deletion priority value from the respective plurality of pieces of content until one of memory of the apparatus has sufficient storage capacity for the at

least one subsequent piece of content, or each of the respective plurality of pieces of content has been identified and deleted.

16. (Previously Presented) The apparatus of Claim 15, wherein, when the other apparatus determines a plurality of pieces of content have an exceeded client expiration time, and when the memory does not have sufficient storage capacity for at least one subsequent piece of content and each of the respective plurality of pieces of content has been identified and deleted, receiving one or more instructions comprises receiving one or more instructions to delete at least one piece of content having a highest deletion priority value from any pieces of content remaining in memory of the apparatus, the at least one piece of content having been identified by the other apparatus as the piece of content having the highest deletion priority value from a comparison between the deletion priority values of the pieces of content remaining in memory of the apparatus.

17. (Previously Presented) The apparatus of Claim 12, wherein the memory stores executable instructions that in response to execution by the processor cause the apparatus to further perform the following:

associating each piece of content stored in the memory with respective parameters.

18. (Previously Presented) The apparatus of Claim 17, wherein the memory stores executable instructions that in response to execution by the processor cause the apparatus to further perform the following:

setting the deletion priority value for at least one piece of content.

19. (Previously Presented) A method of controlling storage of content in memory, the method comprising:

receiving, at a network entity from a terminal located remote from the network entity, a status of at least one piece of content stored in memory of the terminal, wherein each piece of

content is associated with parameters including a client expiration time and a deletion priority value; and

sending one or more instructions from the network entity to the terminal based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control, from the network entity, storage of content in memory of the terminal.

20. (Previously Presented) The method of Claim 19, wherein sending one or more instructions comprises:

determining when memory of the terminal has sufficient storage capacity for at least one subsequent piece of content; and when memory does not have sufficient storage capacity,

sending one or more instructions to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory of the terminal.

21. (Previously Presented) The method of Claim 20, wherein sending one or more instructions to delete at least one piece of content comprises:

determining a plurality of pieces of content having an exceeded client expiration time; and

identifying, and thereafter sending one or more instructions to delete, a piece of content having a highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, the comparison excluding any piece of content without an exceeded client expiration time.

22. (Previously Presented) The method of Claim 21, wherein identifying, and thereafter sending one or more instructions to delete, a piece of content comprise repeatedly identifying, and thereafter sending one or more instructions to delete, a piece of content until one of memory of the terminal has sufficient storage capacity for the at least one subsequent piece of

content, or each piece of content having an exceeded client expiration time has been identified and deleted.

23. (Previously Presented) The method of Claim 22, wherein when memory of the terminal does not have sufficient storage capacity for at least one subsequent piece of content and each piece of content having an exceeded client expiration time has been identified and deleted, the method further comprises:

identifying, and thereafter sending one or more instructions to delete, a piece of content having a highest deletion priority value from a comparison between the deletion priority values of any pieces of content remaining in memory of the terminal.

24. (Previously Presented) The method of Claim 19 further comprising:

receiving at least one piece of content at the network entity; and

sending at least one piece of content to the terminal such that the terminal receives, and thereafter stores, the at least one piece of content sent thereto.

25. (Previously Presented) The method of Claim 24, wherein the parameters further include a server expiration time, and wherein the method further comprises:

monitoring the server expiration time of the at least one piece of content in memory of the network entity to determine when at least one piece of content has an exceeded server expiration time; and when at least one piece of content has an exceeded server expiration time,

deleting the at least one piece of content having an expired server expiration time.

26. (Previously Presented) The method of Claim 19 further comprising:

associating each piece of content stored in memory of the terminal with respective parameters.

27. (Previously Presented) The method of Claim 26, wherein associating each piece of content comprises setting the deletion priority value for at least one piece of content at the terminal.

28. (Previously Presented) The method of Claim 26, wherein associating each piece of content comprises associating each piece of content stored in memory of the terminal with respective parameters at the network entity.

29. (Previously Presented) A computer program product for controlling storage of content in memory, the computer program product comprising a computer-readable storage medium having computer-readable program code portions stored therein that in response to execution by a processor cause an apparatus to at least perform the following:

receiving, from a terminal located remote from the the apparatus, a status of at least one piece of content stored in memory of the terminal, wherein each piece of content is associated with parameters including a client expiration time and a deletion priority value; and

sending one or more instructions from the apparatus to the terminal based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control, from the apparatus, storage of content in memory of the terminal.

30. (Previously Presented) The computer program product of Claim 29, wherein sending one or more instructions comprises:

determining when memory of the terminal has sufficient storage capacity for at least one subsequent piece of content; and when memory does not have sufficient storage capacity,

sending one or more instructions to instruct at least one of the terminal or a user of the terminal to delete at least one piece of content based upon a comparison between the deletion priority values of a plurality of pieces of content stored in memory of the terminal.

31. (Previously Presented) The computer program product of Claim 30, wherein sending one or more instructions to delete at least one piece of content comprises:

determining a plurality of pieces of content having an exceeded client expiration time;

identifying a piece of content having a highest deletion priority value from a comparison between the deletion priority values of the pieces of content having an exceeded client expiration time, the comparison excluding any piece of content without an exceeded client expiration time; and

sending one or more instructions to instruct the terminal to delete the identified piece of content.

32. (Previously Presented) The computer program product of Claim 31, wherein identifying a piece of content, and sending one or more instructions to instruct the terminal to delete the identified piece of content, comprise repeatedly identifying a piece of content, and sending one or more instructions to instruct the terminal to delete the identified piece of content, until one of memory of the terminal has sufficient storage capacity for the at least one subsequent piece of content, or each piece of content having an exceeded client expiration time has been identified and deleted.

33. (Previously Presented) The computer program product of Claim 32, wherein the computer-readable storage medium has computer-readable program code portions stored therein that in response to execution by the processor, when memory of the terminal does not have sufficient storage capacity for at least one subsequent piece of content and each piece of content having an exceeded client expiration time has been identified and deleted, cause an apparatus to further perform the following:

identifying, and thereafter sending one or more instructions to instruct the terminal to delete, a piece of content having a highest deletion priority value from a comparison between the deletion priority values of any pieces of content remaining in memory of the terminal.

34. (Previously Presented) The computer program product of Claim 30, wherein the computer-readable storage medium has computer-readable program code portions stored therein

that in response to execution by the processor cause the apparatus to further perform the following:

receiving at least one piece of content at the apparatus; and

sending at least one piece of content to the terminal such that the terminal receives, and thereafter stores, the at least one piece of content.

35. (Previously Presented) The computer program product of Claim 34, wherein the parameters further include a server expiration time, and wherein the computer-readable storage medium has computer-readable program code portions stored therein that in response to execution by the processor cause the apparatus to further perform the following:

monitoring the server expiration time of the at least one piece of content in memory of the apparatus to determine when at least one piece of content has an exceeded server expiration time; and when at least one piece of content has an exceeded server expiration time,

deleting the at least one piece of content having an expired server expiration time.

36. (Previously Presented) The computer program product of Claim 29, wherein the computer-readable storage medium has computer-readable program code portions stored therein that in response to execution by the processor cause the apparatus to further perform the following:

associating each piece of content stored in memory of the terminal with respective parameters.

37. (Previously Presented) The computer program product of Claim 36, wherein the computer-readable storage medium has computer-readable program code portions stored therein that in response to execution by the processor cause the apparatus to further perform the following:

setting the deletion priority value for at least one piece of content at the terminal.

38. (Previously Presented) The computer program product of Claim 36, wherein associating each piece of content comprises associating each piece of content stored in memory of the terminal with respective parameters at the apparatus.

39. (Previously Presented) An apparatus comprising:

a means for storing at least one piece of content, wherein each piece of content is associated with parameters including a client expiration time and a deletion priority value;

a means for sending a status of the at least one piece of content stored by the apparatus to a network entity located remote from the apparatus; and

a means for receiving one or more instructions from the network entity based upon the status and the associated parameters, including the client expiration time and deletion priority value, to at least partially control storage of the at least one piece of content by the apparatus.

9. ***Evidence Appendix.***

None.

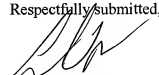
10. ***Related Proceedings Appendix.***

None.

CONCLUSION

For at least the foregoing reasons, Appellants respectfully request that the rejections be reversed.

Respectfully submitted,



Andrew T. Spence
Registration No. 45,699

CUSTOMER NO. 00826
ALSTON & BIRD LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111
LEGAL02/31838166v1